WHAT IS CLAIMED IS:

1. A machine-implemented method comprising:

receiving, by a first process, a shortcut to a physical address associated with a level of a multi-level virtual address translation table;

posting a descriptor comprising a virtual address and a shortcut to an interface between the first process and a second process; and

determining the physical address corresponding to the virtual address based on at least the virtual address and the shortcut.

2. The method of claim 1 further comprising transferring data to or from the buffer located at the physical address.

3. The method of claim 1 further comprising:

generating the shortcut by a third process.

4. The method of claim 3 wherein generating the shortcut by the third process comprises:

receiving a request to register a virtual buffer, the request including a virtual address corresponding to the start of the virtual buffer;

determining the physical address of one level of the multi-level address translation table associated with the virtual memory space in which the virtual buffer resides; and

generating a shortcut based on the physical address of the one level of the multi-level address translation table.

5. The method of claim 4 wherein generating a shortcut further comprises:

generating the shortcut based on a key unknown to the first process.

6. The method of claim 4 wherein generating a shortcut further comprises:

generating the shortcut based on a function unknown to the first process.

7. The method of claim 1 further comprising:

retrieving a key by the second process; and

applying the key to the shortcut to produce the physical address associated with one level of a multi-level virtual address translation table.

8. The method of claim 1 further comprising determining if the physical address is associated with the first address.

9. The method of claim 1 further comprising determining if the virtual page containing the virtual address is pinned into physical memory.

10. The method of claim 1 wherein the interface is a virtual interface.

11. The method of claim 1 further comprising determining if the first process is authorized to access the virtual address.

12. The method of claim 1 further comprising determining if descriptors posted to the interface between the first process and second process are authorized to access the virtual address.

13. The method of claim 1 further comprising:

receiving, by a first process, a plurality of shortcuts, each shortcut to a physical address associated with a level of a multi-level virtual address translation table.

14. The method of claim 4 wherein generating a shortcut comprises:

applying a function, F, to the physical address of the one level and a key.

15. The method of claim 14 wherein the key is associated with the interface between the first and second process.

16. The method of claim 14 wherein the key is associated with the first process.

17. A machine-implemented method comprising:

generating, by a first process, a request to register a virtual buffer mapped to physical memory by a multi-level virtual address translation table associated with the first process;

determining a block of memory that includes the physical address corresponding to the start of the virtual buffer; and

generating, by a second process, one or more shortcuts based on the block of memory including the physical address corresponding to the start of the virtual buffer.

18. The method of claim 17 wherein generating a shortcut further comprises:

generating the shortcut based on a key, which is unknown to the first process.

19. The method of claim 17 wherein generating a shortcut further comprises:

generating the shortcut based on a function, which is unknown to the first process.

20. The method of claim 17 further comprising:

transmitting a request to a third process to perform an input or output operation on the virtual buffer, wherein the request includes the shortcut and a virtual address associated with the virtual buffer; and

determining a physical address of the virtual address based on the virtual address and the shortcut.

21. The method of claim 20 further comprising:

determining if the physical address is associated with the first address; and

if the physical address is associated with the first address, then enabling the input or output operation on at least part of the virtual buffer.

22. The method of claim 20 further comprising:

determining if the associated physical pages are pinned into physical memory; and

if the associated virtual pages are pinned into physical memory, then enabling the input or output operation on at least part of the virtual buffer.

23. The method of claim 20 further comprising:

determining if the requesting process is authorized to access the associated virtual buffer; and if the requesting process is authorized to access the associated virtual buffer, then enabling the input or output operation on at least part of the virtual buffer.

24. The method of claim 20 further comprising:

determining if requests posted to the interface between the first process and the third process are authorized to access the associated virtual buffer; and

if requests to the interface are authorized to access the associated virtual buffer, then enabling the input or output operation on at least part of the virtual buffer.

25. The method of claim 18 further comprising:

transmitting a request to a third process to perform an input or output operation on the virtual buffer, wherein the request includes one of the one or more shortcuts and a virtual address associated with the virtual buffer; and

determining a physical address of the virtual address based on the virtual address, the shortcut and the key.

26. A system comprising:

a first processor capable of:

executing instructions of a first process which causes the first processor to produce a shortcut to a physical address associated with a level of a multi-level virtual address translation table; and

executing instructions of a second process which causes the first processor to post a descriptor comprising a virtual address and the shortcut to an interface; and

a second processor capable of executing instructions of a third process which cause the second processor to:

read the descriptor posted on the interface; and determine a physical address of the virtual address based on at least the virtual address and the shortcut.

27. The system of claim 26 wherein the instructions of the first process cause the first processor to encrypt the shortcut with a key.

28. The system of claim 27 wherein the instructions of the third process cause the second processor to:

retrieve the key; and

apply the key to the shortcut to produce the physical address associated with one level of a multi-level virtual address translation table.

29. The system of claim 28 wherein the instructions of the third process cause the second processor to determine if the physical address is associated with the second process.

30. The system of claim 28 wherein the instructions of the third process cause the second processor to determine if the associated virtual pages are pinned into physical memory.

31. The system of claim 28 wherein the instructions of the third process cause the second processor to determine if the second process is authorized access to the virtual buffer.

32. The system of claim 27 wherein the instructions of the third process cause the second processor to determine if requests posted to the interface between the second process and the third process are authorized access to the virtual buffer.

33. A computer program product residing on a computer readable medium having instructions stored thereon that, when executed by the processor, cause that processor to:

produce a shortcut to a physical address associated with a level of a multi-level virtual address translation table; and

write a descriptor comprising a virtual address and the shortcut to an interface.

34. The product of claim 33 having instructions that further cause the processor to encrypt the shortcut with a key.

35. The product of claim 33 having instructions that further cause the processor to encrypt the shortcut with a function.

36. A computer program product residing on a computer readable medium having instructions stored thereon that, when executed by the processor, cause that processor to:

read a message posted on an interface by a first process, the message including a shortcut to a physical address associated with a level of a multi-level virtual address translation table; and

determine a physical address of the virtual address based on at least the virtual address and the shortcut.

37. The product of claim 36 having instructions that further cause the processor to:

retrieve a key; and

apply the key to the shortcut to produce the physical address associated with one level of a multi-level virtual address translation table.

38. The product of claim 36 having instructions that further cause the processor to determine if the physical address is associated with the first process.

39. The product of claim 36 having instructions that further cause the processor to determine if the virtual pages referenced by the message are pinned in physical memory.

40. The product of claim 36 having instructions that further cause the processor to determine if the first process is authorized access to the virtual buffer referenced by the message.

41. The product of claim 36 having instructions that further cause the processor to determine if messages posted on the interface are authorized access to the virtual buffer.

42. A system comprising:

a client computer; and

a server in communication with the client computer using a network, the server comprising:

a first processor capable of producing a shortcut to a physical address associated with a level of a multi-level virtual address translation table and writing a descriptor comprising a virtual address and the shortcut to an interface; and

a second processor capable of reading the descriptor posted on the interface, determining a physical address of the virtual address based on at least the virtual address and the shortcut and transferring data located at the physical address to the client computer using the network.

43. The system of claim 42 wherein the first processor is capable of encrypting the shortcut with a key.

44. The system of claim 43 wherein second processor is capable of decrypting the shortcut to produce the physical address associated with one level of a multi-level virtual address translation table.

45. The system of claim 42 wherein the interface is a virtual interface.

46. A system comprising:

a storage device; and

a server in communication with the storage computer using a network, the server comprising:

a first processor capable of producing a shortcut to a physical address associated with a level of a multi-level virtual address translation table and writing a descriptor comprising a virtual address and the shortcut to an interface; and

a second processor capable of reading the descriptor posted on the interface, determining a physical address of the virtual address based on at least the virtual address and the shortcut and transferring data located at the physical address to the storage device using the network.

47. The system of claim 46 wherein the first processor is capable of encrypting the shortcut with a key.

48. The system of claim 47 wherein second processor is capable of decrypting the shortcut to produce the physical address associated with one level of a multi-level virtual address translation table.

49. The system of claim 46 wherein the interface is a virtual interface.